JOINT STRIKE FIGHTER

'Joint Strike Fighter', 'JSF', and the JSF Logo are Trademarks of the United States Government.

Open Systems Architecture For JSF Avionics

30 April 1998

Maj Dan Vore

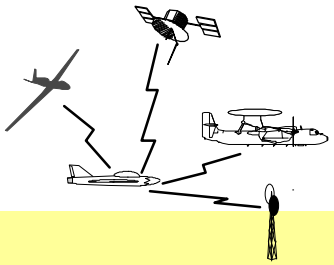
Integrated Core Processor / Open Systems Architecture Lead

TOPICS

- Context For JSF Avionics Architecture
- Open System Principles for Embedded Real Time Processing
- JSF Avionics Architecture Definition (JAAD)
- System of Systems (SOS)
- Technology Obsolescence
- Summary

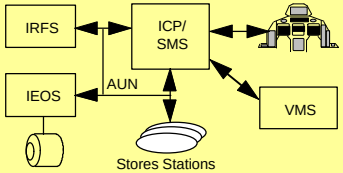
DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE: *ARCHITECTURAL HIERARCHY*

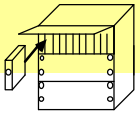


Level 1: Force Structure/ System of Systems

Level 2: Weapon System/ JSF



Level 3: Major Subsystem/ Avionics Suite



Level 4: Functional Area/ Integrated Core Processing

Level 5: Hardware/Software Building Block

Level 6: Hardware/Software Component

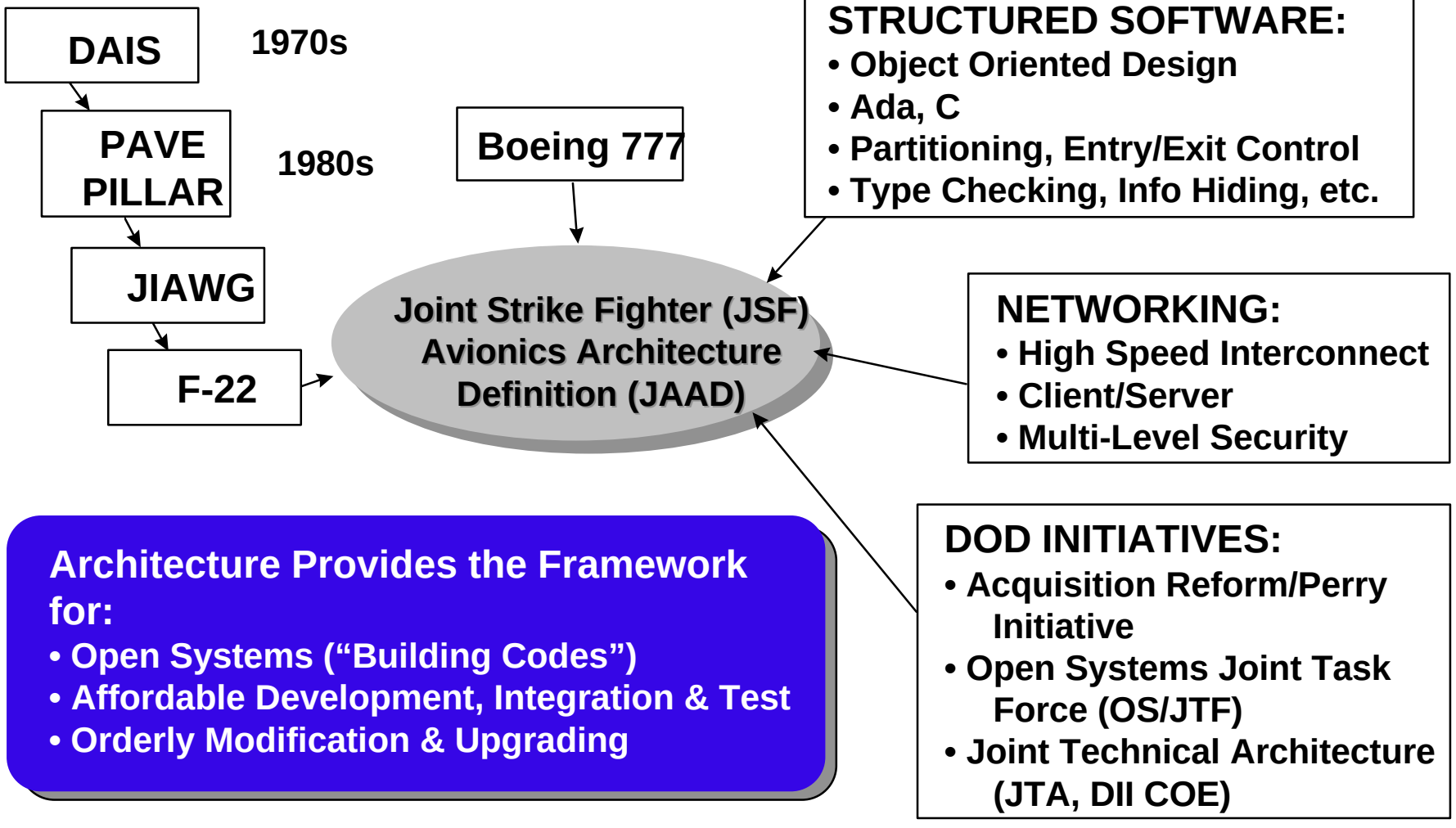
DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE: *NATURE OF MODERN AVIONICS ARCHITECTURES*

- **Any System Architecture Must Address:**
  - **Functional Entities and Their Interfaces**
  - **Interconnects/Interaction Among Entities**
  - **Global Design Rules & System Attributes**
- **Open Systems Principles Provide Discipline For:**
  - **Functional & Physical Partitioning**
  - **Definition & Control of Interfaces & System Services**
  - **Appropriate Use of Standards (Preferably Commercial)**
- **Characteristics Include:**
  - **High Performance, Real Time Embedded Processing**
  - **Scalability**
  - **Technology Independence & Domain-Confined Timing**
  - **Information Security**
  - **Reliability, Maintainability & Fault Management**
  - **Partitioning for Affordable Upgrading & Testing:**
    - **Isolated Threads**
    - **Modular Packaging**
    - **OS Kernel**
    - **Memory Partitioning (page tables)**
    - **ORBs**
    - **Input Parameter Checking**

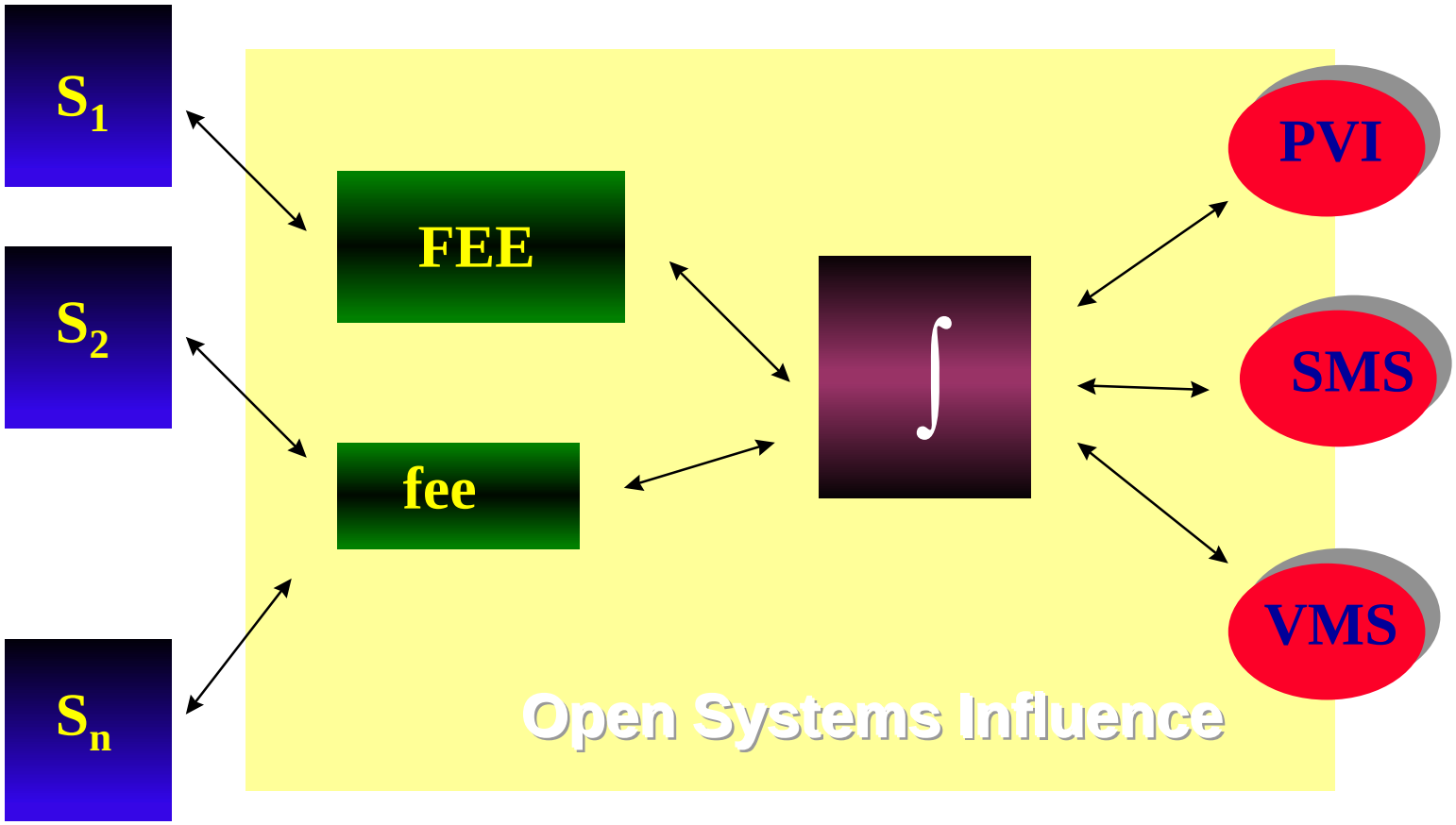
DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE:
LEGACY OF MODERN FIGHTER AVIONICS



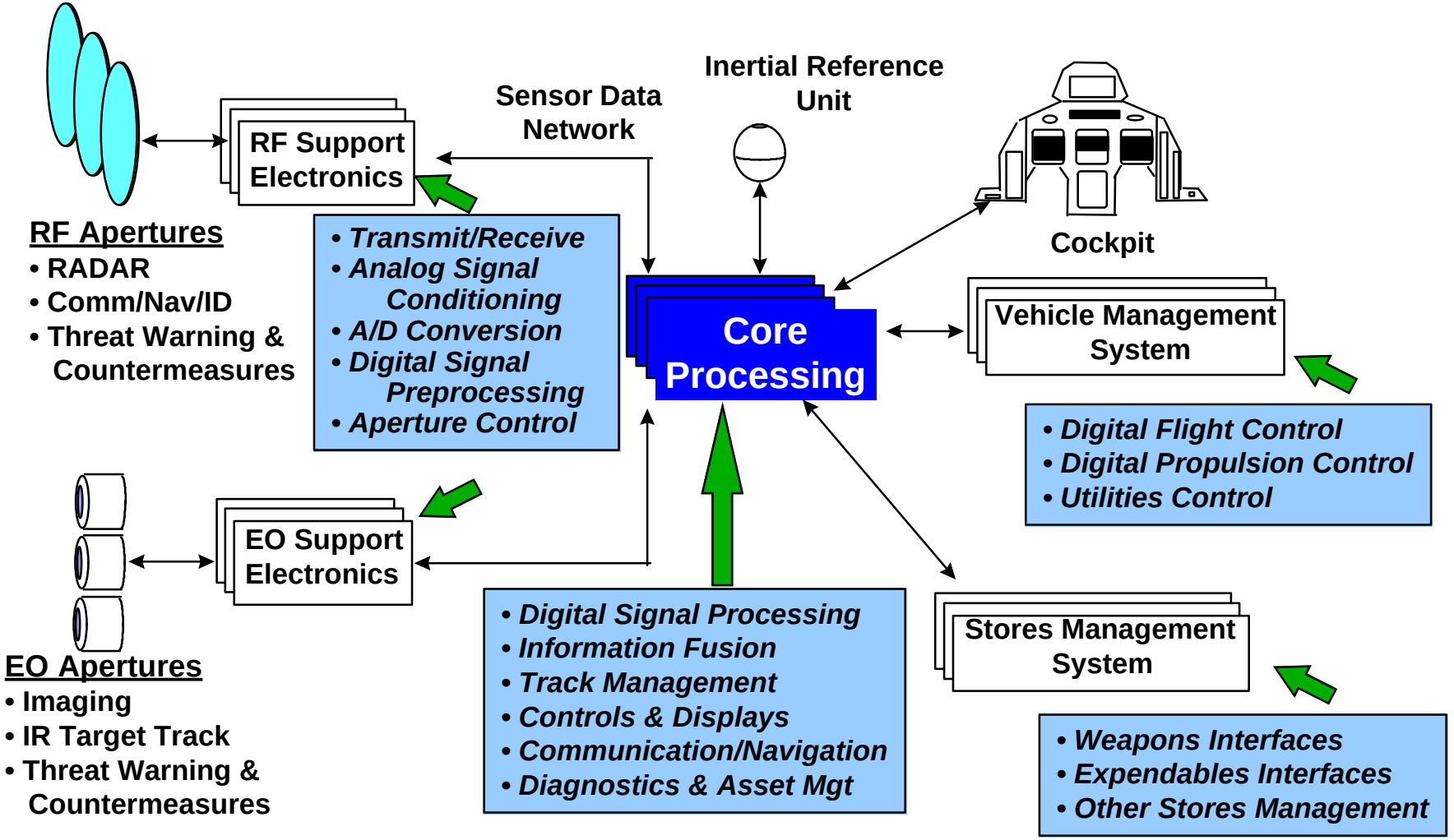
DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE:
GENERIC FIGHTER AVIONICS ARCHITECTURE



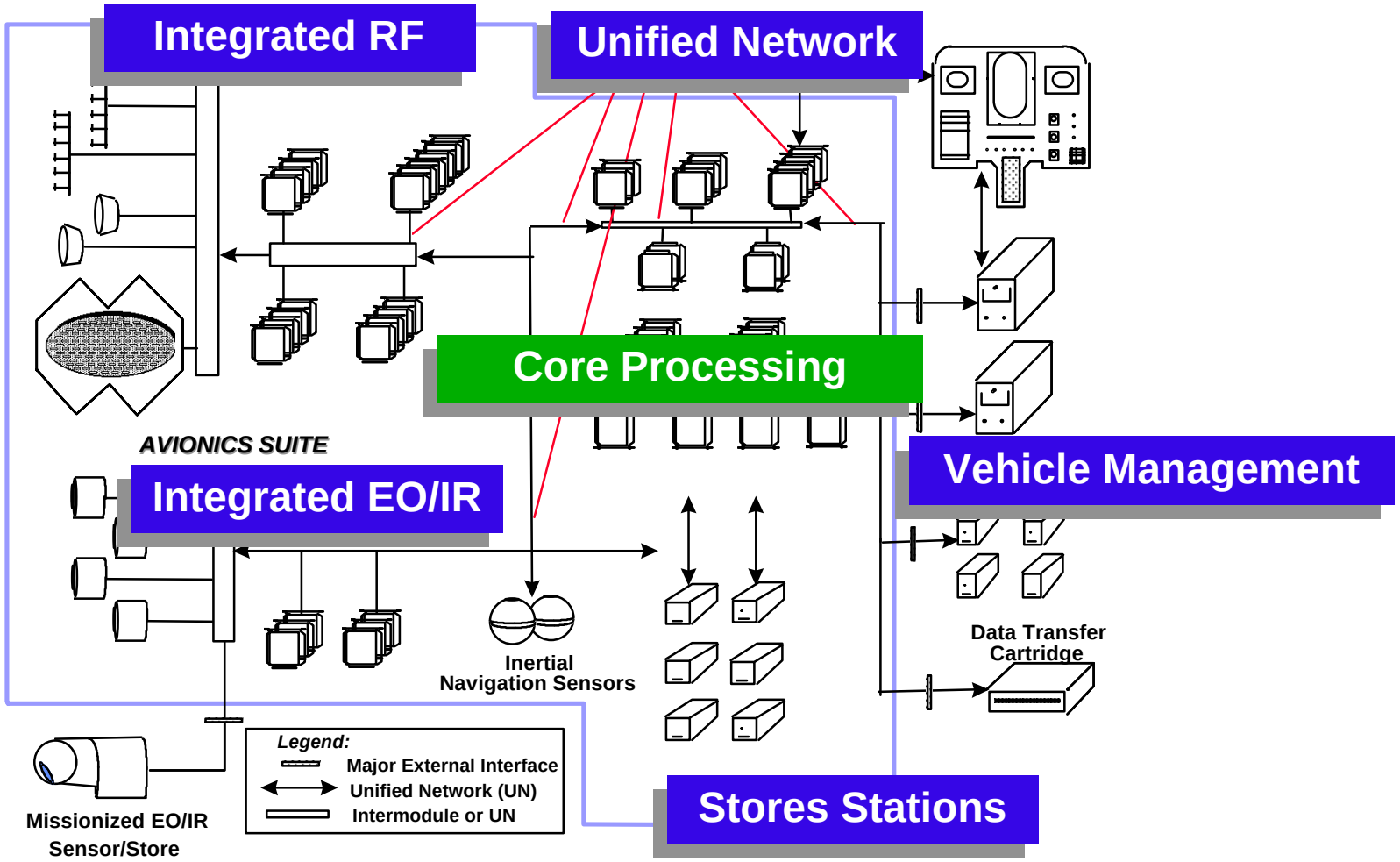
DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE: *STRUCTURE OF A MODERN FIGHTER AVIONICS SUITE*



DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE:
JSF LEVEL 3 PARTITIONING



DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE: *LESSONS LEARNED*

- **Open Systems Is Necessary but Not Sufficient to Achieve Long-term:**
  - **Performance & Support**
  - **Affordability**
  - **Interoperability With DoD**
  - **System of Systems**
- **Besides Open Systems, Program Strategy Must Enforce:**
  - **Software Engineering Discipline (i.e., S/SEE)**
  - **Rigorous Requirements Analysis**
  - **Contract Incentives to Optimize *System* Performance**
- **Functional Integration Must Only Occur At Selected Junctures Within the Architecture (e.g., Track Fusion Node) -- Physical Integration + Functional Federation**

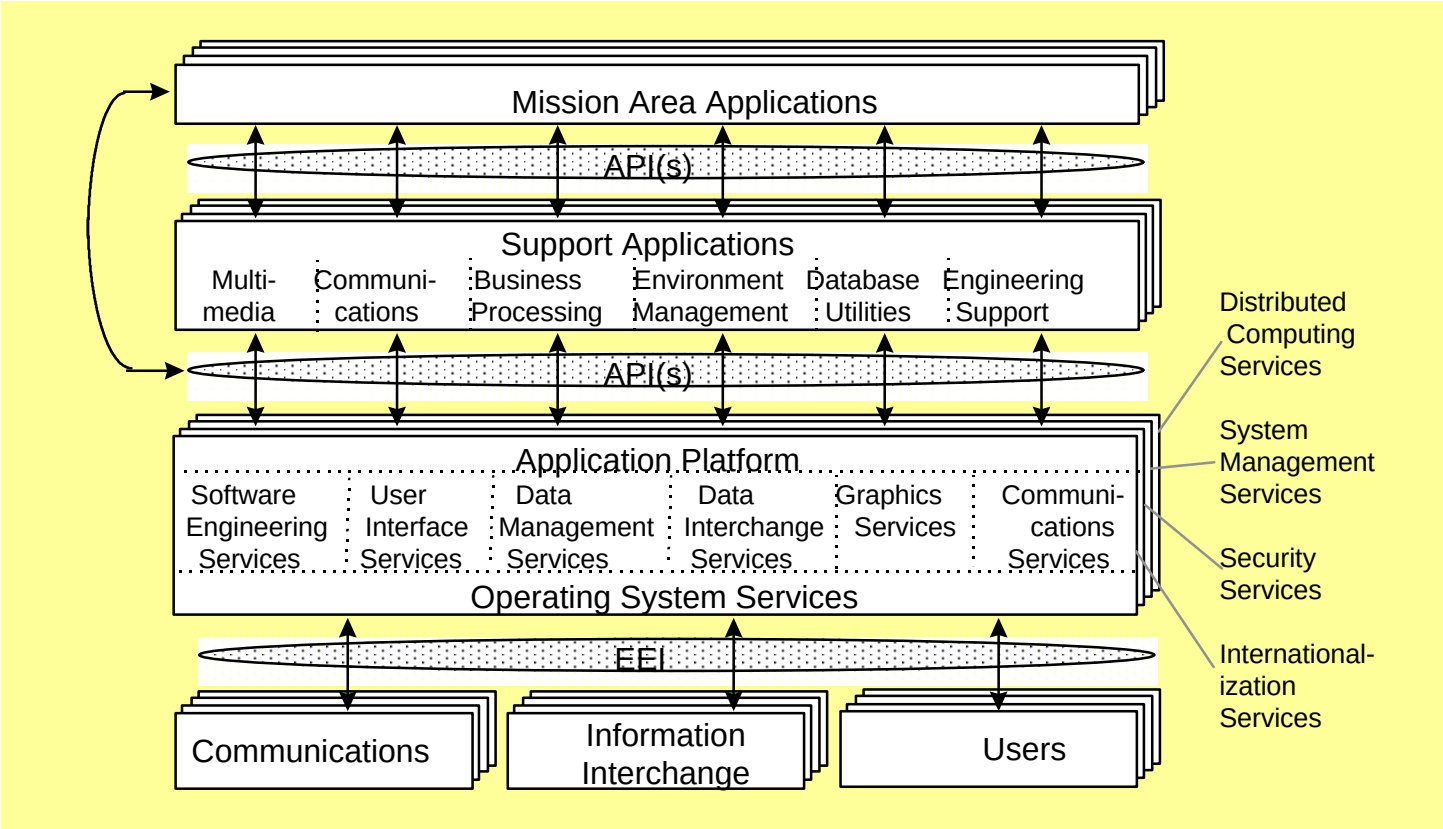
DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE: *LESSONS LEARNED* (CONTINUED)

- **Employ Open Systems Principles at the Outset**
- **Place Primary Emphasis on Interfaces & Common Services, *Not* Common Modules**
- **Adequate Risk Reduction and Demonstration of Candidate Technologies Before Standards Selection**
- **C^4 and Avionics Systems Face Many of the Same Open Systems Challenges -- Issues Addressed In Architectures May Differ**
- **A Technical Reference Model (TRM) is a Powerful Complement to Traditional Architecture Definitions**

DISTRIBUTION A. Approved for public release; distribution

CONTEXT FOR JSF AVIONICS ARCHITECTURE:
DOD TRM, VERSION 3.0



- **A Key Underpinning of the JAAD**

DISTRIBUTION A. Approved for public release; distribution

TOPICS

- Context For JSF Avionics Architecture
- Open System Principles for Embedded Real Time Processing
- JSF Avionics Architecture Definition (JAAD)
- System of Systems
- Technology Obsolescence
- Summary

DISTRIBUTION A. Approved for public release; distribution

OPEN SYSTEM PRINCIPLES:
SOME KEY TERMS & CONCEPTS

- **Classes of Information Systems:**
  - **Management Information Systems:**
    - **Payroll, Purchasing, Personnel, etc. - All COTS**
    - **Large, Physically Dispersed, Non-Real Time**
    - **Classic Opportunity for Open Systems - Competitive Sourcing, Software Reuse, System Interoperability, etc.**
  - **Command, Control, Communications & Intelligence (C^3I):**
    - **Distributed/Parallel High Speed Processing w/ Mix of COTS & Custom Components**
    - **Operate in Non- or Soft Real Time**
    - **Major Benefits from Open Systems**
  - **Embedded:**
    - **Small, High Performance, Single Platform - COTS May be Limited**
    - **Real Time Operation - *HARD* Real Time for Flight & Mission-Critical Functions**
    - **Open Systems Strategy Must Meet Performance, Safety, Interoperability, & Other Requirements**

DISTRIBUTION A. Approved for public release; distribution

OPEN SYSTEM PRINCIPLES: *SOME KEY TERMS & CONCEPTS* (CONTINUED)

- **Integration:**
  - **Physical Integration - Collocation/Interconnection of System Elements**
  - **Functional Integration - Information Sharing, Cooperative Functions, Redundancy, Centralized Resource Management, etc.**
- **Modularity:**
  - **Partitioning/Decomposition of System Resources into "Atomic" Building Blocks**
  - **Interfaces Among Modules at Various Levels of the System Hierarchy**
- **Commonality: Use of the Same Building Blocks Within and Among Systems**

DISTRIBUTION A. Approved for public release; distribution

OPEN SYSTEM PRINCIPLES: *GENERAL ATTRIBUTES OF OPEN SYSTEMS* (CONTINUED)

- **Modular Design w/Mapping of Functions Onto Hardware & Software Components**
- **Mapping of Software Architecture Onto Hardware Architecture (a Block Diagram Is *NOT* an Architecture)**
- **Component Interfaces Which Are:**
  - **Fully Defined**
  - **Publicly Available**
  - **Maintained Through a Process of Expert Consensus**
  - **Implementable with Available Products**
- **Maximum Feasible Use of Mature, Well Supported, Widely Used Interface Standards**
- **Litmus Test Is Ability to Mix/Match/Integrate Modules from Multiple Sources**

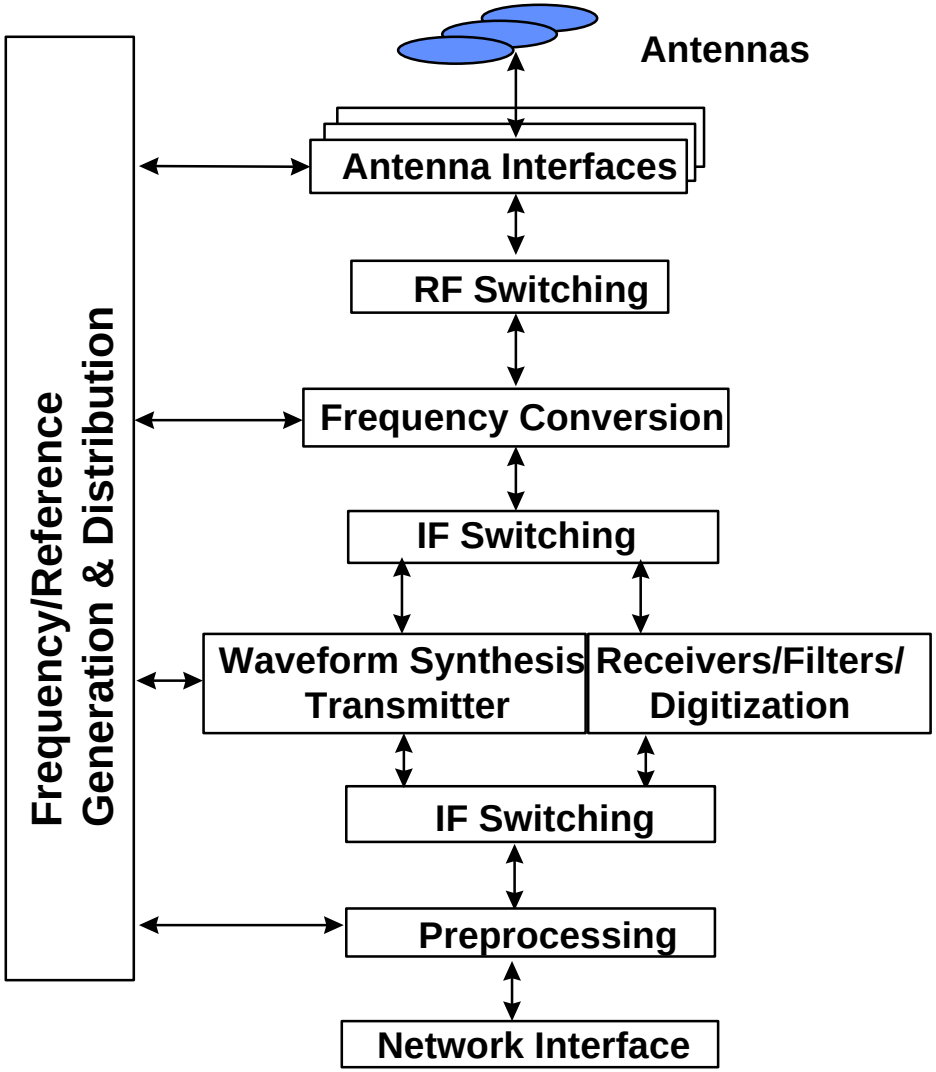
DISTRIBUTION A. Approved for public release; distribution

OPEN SYSTEM PRINCIPLES:
KEY AVIONICS ARCHITECTURE ISSUES AND TRADE-OFFS

- **Interconnects:**
  - **Unified Avionics Network Protocol**
  - **Required Performance to Support Information Flows**
- **Memory-Mapped vs. Message Passing Data Communications**
- **Integrated vs. Distributed Data & Signal Processing**
- **Processor Types (GP vs. SP Processors)**
- **Integrated vs. Single Purpose Apertures**
- **Real Time Operating System (RTOS) Characteristics & Software Building Codes (e.g., APIs)**
- **Software Language & Development Tools**
- **Isolated Functional Threads vs. Physically Federated Subsystems**

DISTRIBUTION A. Approved for public release; distribution

OPEN SYSTEM PRINCIPLES:
OPEN SYSTEMS APPLIES TO NON-DIGITAL FUNCTIONS



- **Partitioning & Interface Definitions**
- **Re-use of Common Building Blocks**
- **Selective Tech Insertion**
- **Competitive Sourcing**

DISTRIBUTION A. Approved for public release; distribution

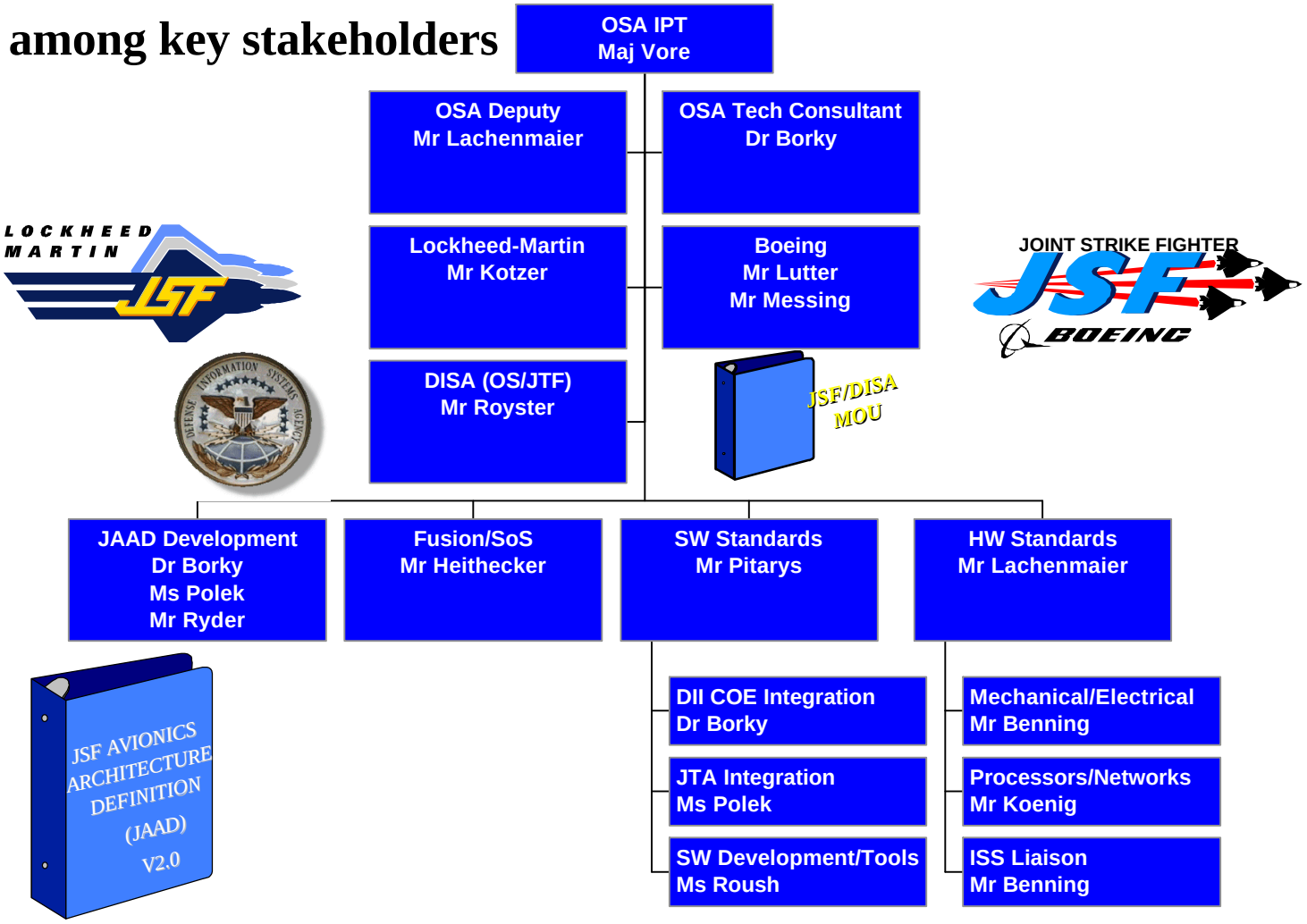
TOPICS

- Context For JSF Avionics Architecture
- Open System Principles for Embedded Real Time Processing
- JSF Avionics Architecture Definition (JAAD)
- System of Systems (SOS)
- Technology Obsolescence
- Summary

DISTRIBUTION A. Approved for public release; distribution

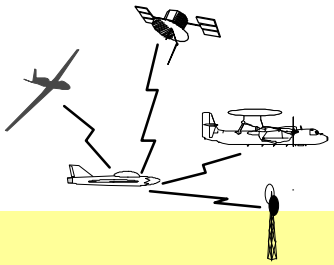
JAAD:
JSF OSA IPT ORGANIZATION

- **Teamwork among key stakeholders**



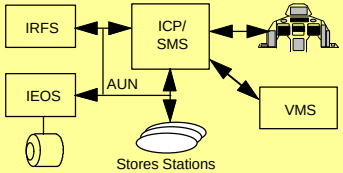
DISTRIBUTION A. Approved for public release; distribution

JSF AVIONICS ARCHITECTURE DEFINITION (JAAD): *ARCHITECTURAL HIERARCHY*

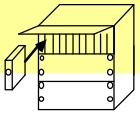


Level 1: Force Structure/ System of Systems

Level 2: Weapon System/ JSF



Level 3: Major Subsystem/ Avionics Suite



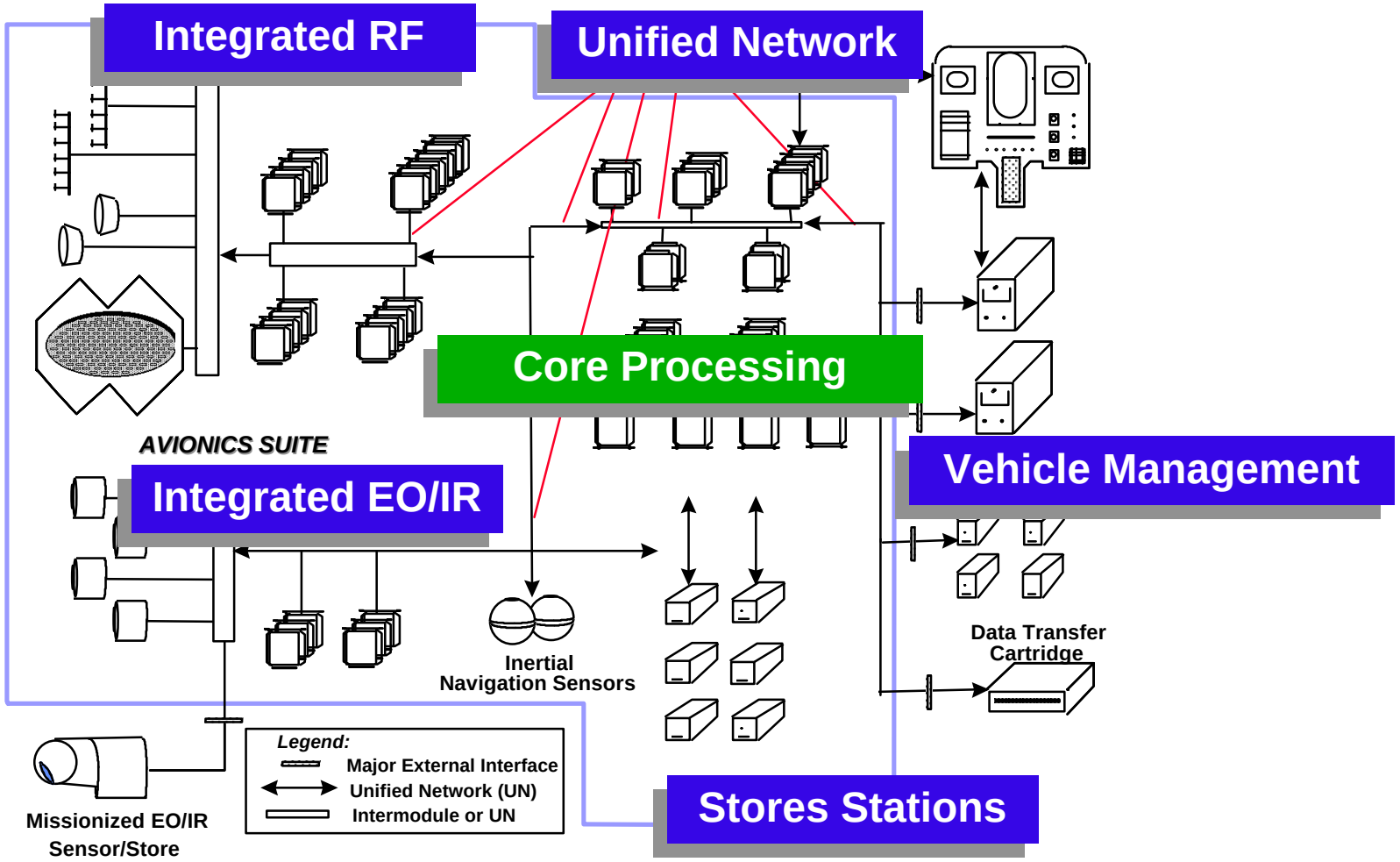
Level 4: Functional Area/ Integrated Core Processing

Level 5: Hardware/Software Building Block

Level 6: Hardware/Software Component

DISTRIBUTION A. Approved for public release; distribution

JAAD:
JSF LEVEL 3 PARTITIONING



DISTRIBUTION A. Approved for public release; distribution

JAAD:
PRIMARY ISSUES FOR JSF ARCHITECTURE & ACQ STRATEGY

- **Cost of Module/Backplane Redesign For Tech Insertion**
  - **Module Design and Packaging Must Preserve COTS Benefits**
  - **Backplane Design Must Ensure Multi-generational Processor Replacements Within COTS Product Lines**
  -
- **Cost and Timeliness of Software Changes**
  - **Software Architecture Must Allow For Feature Insertion Or Extraction Within a Single Baseline OFP**
  - **Engineering Environment Must Merge and Automate Intermediate Steps Between Development and System-level Validation**
  - **Must Establish and Maintain Boundaries Around, and Eliminate Dependencies Among, Functions**
  - **Must Reduce Regression, Validation and Flight Test Times**

DISTRIBUTION A. Approved for public release; distribution

JAAD:
PRIMARY ISSUES FOR JSF ARCHITECTURE & ACQ STRATEGY (CONT'D)

- **Cannot Afford Two Distinct Module Product Lines**
  - **Tech Insertion Must Integrate Fully With *Both* Production and Sustainment**
  - **R&M Improvements From Tech Insertion Must Be Reflected In Sustainment**

- **Cost Prohibitive to Port Among Differing Engineering & Support Environments**
  - **Single, Integrated S/SEE Across Contractor Team**
  - **Open Architecture Must Permeate System Infrastructure**
  - **Support Facilities Must Mirror Modular System Design**

DISTRIBUTION A. Approved for public release; distribution

JAAD:
CHARACTERISTICS

- ***Spans Technical and System Architectures***
- ***Practical* Openness - Emphasis on Integration & Upgrading**
- ***Scalability* - Achieved by Modular Design & Overall Architecture**
- ***Domain Confined Timing* - Strict Localization of Critical Timing Relationships**
- ***Failure Management* - Embedded Diagnostics/Prognostics Plus Selective Redundancy**
- ***Unified Network* - Simplify Integration & Modification; Reduce Cost & Weight**
- ***F^3I* - Where "Function" is Defined as The Set of Behaviors Which a Module Can Exhibit Through Its Interface**
- ***Maximum Feasible Use of COTS & Commercial Standards***
- ***Support for the "ilities"* - Testability, Maintainability, etc.**

DISTRIBUTION A. Approved for public release; distribution

JAAD:
PRIMARY BUILDING CODES

- **Connectivity:**
  - **Unified Network Protocol**
  - **Links to Other Platforms (“System of Systems”)**
- **Software:**
  - **Higher Order Language(s)**
  - **Real Time Operating System (RTOS) Services**
  - **Application Programming Interface(s) (APIs)**
  - **Graphics Interface**
  - **Data Base Management System Interface**
  - **Object Request Brokers (ORBs, if used)**
- **Mechanical/Electrical:**
  - **Packaging & Cooling**
  - **Power Distribution**

DISTRIBUTION A. Approved for public release; distribution

JAAD:
PROVIDES EXAMPLE EVALUATION PROCESS FOR CANDIDATE STANDARDS

- **Initial Candidate Protocols for Unified Avionics Network**
  - **Asynchronous Transfer Mode (ATM)**
  - **Fibre Channel**
  - **Fibre Channel AE**
  - **Gigabit Ethernet**
  - **Myrinet**
  - **Scalable Coherent Interface (SCI)**
  - **SCI/Real Time**
  - **Serial Express**
  - **Sun S-Connect**

DISTRIBUTION A. Approved for public release; distribution

JAAD:
ENABLERS OF KEY ISSUE RESOLUTION

ISSUE

- **Tech Insertion Cost**
  - **Reduce Module Redesign**
  - **Preservation of COTS**

"The contractor shall submit a plan for technology insertion which clearly demonstrates minimal or no cost of module or backplane redesign for incorporation of successive generations of commercial processors."

ENABLER

- **H/W Independence (API/OS)**
- **Modular Partitioning (F^3I)**
- **Executable Specifications**
- **Unified Network/Protocol**
- **On-module Features Reduce Backplane Complexity**
- **Processor Speeds Allow For General Purpose Application and Fewer Modules**
- **Incentivize Prime to Minimize Non-Recurring Cost of Module Design**

DISTRIBUTION A. Approved for public release; distribution

JAAD:
ENABLERS OF KEY ISSUE RESOLUTION

ISSUE	ENABLER
• **Software/OFP Cost and Timelines** – **Single OFP Baseline** – **Responsiveness**	• **Re-Use of Objects and/or Components** • **Use of Structured Decomposition and Object-Oriented Design Approaches** • **Clean Segregation of Applications and Execution Platform via API** • **Replication of Environment Across The Team** • **Independent Funtional Threads and Controlled Fusion Processes** • **Functionality Implemented Via Data Versus Code**

DISTRIBUTION A. Approved for public release; distribution

JAAD:
ENABLERS OF KEY ISSUE RESOLUTION

ISSUE	ENABLER
• Separate Module Production and Sustainment Lines	• Modular Architecture • HW Independence • Use Production Line For Sustainment
• Separate Engineering and Support Environments	• Portable SW and S/SEE • COTS OS, Processors and Networks Allow System Emulation On Development Stations

DISTRIBUTION A. Approved for public release; distribution

JAAD:
ISSUES IN WORK FOR VERSION 3

- **Impact of & Compatibility with DoD Architecture Initiatives:**
  - **RT DII COE, JTA, GOA, AITS RA, etc., etc.**
  - **Overall Use of Object Request Broker (ORB) Construct**
  - **Common Operational Picture - NRO, DARO, NIMA, NSA, etc.**
- **Real Time Attributes - Possible Input to RT DII COE**
- **SOS Support Features:**
  - **Technical Complement to C^4ISR Support Plan**
  - **Definition of SOS from JSF Architectural Viewpoint**
  - **Possible Use of, Input to JTA**
- **Multi-Level Security (MLS)**
- **Prognostics & Health Management (PHM)**
- **Clear Linkage of Architecture To Issue Resolution**

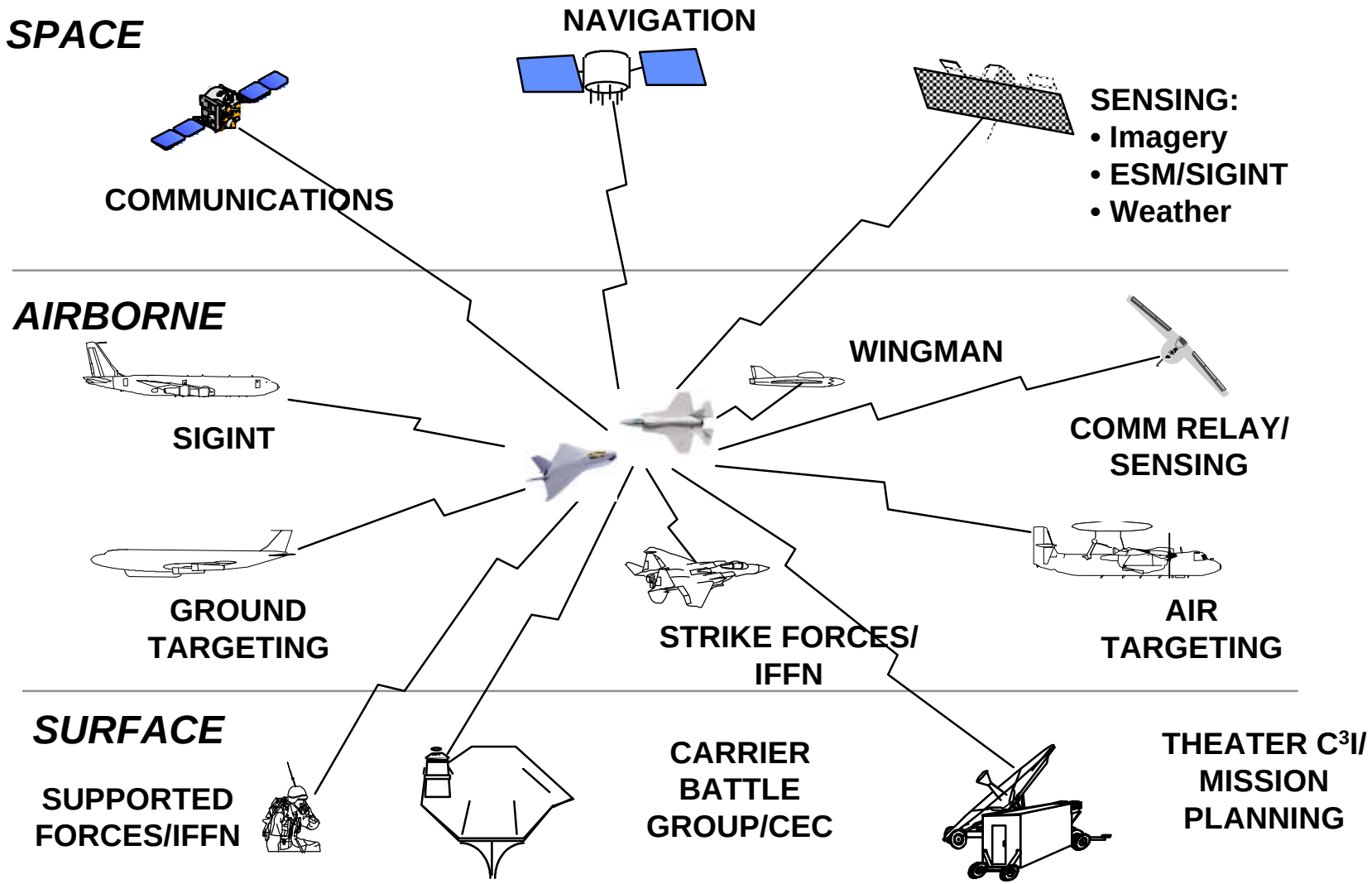
DISTRIBUTION A. Approved for public release; distribution

TOPICS

- Context For JSF Avionics Architecture
- Open System Principles for Embedded Real Time Processing
- JSF Avionics Architecture Definition (JAAD)
- System of Systems (SOS)
- Technology Obsolescence
- Summary

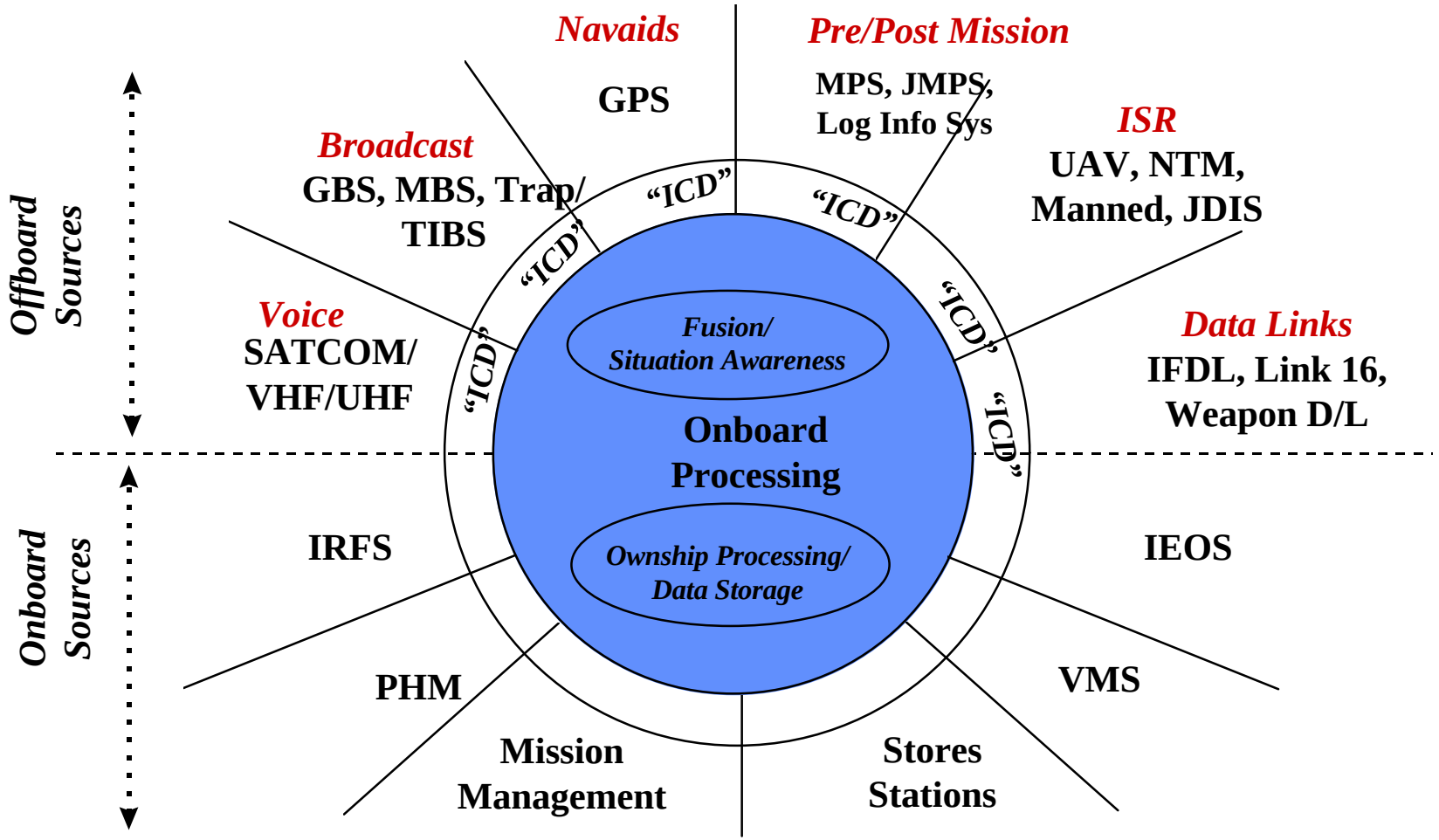
DISTRIBUTION A. Approved for public release; distribution

SOS:
CONTEXT FOR JSF



DISTRIBUTION A. Approved for public release; distribution

SOS:
EXTENDING THE BUILDING CODES FOR JSF OPEN SYSTEM ARCHITECTURE



DISTRIBUTION A. Approved for public release; distribution

TOPICS

- Context For JSF Avionics Architecture
- Open System Principles for Embedded Real Time Processing
- JSF Avionics Architecture Definition (JAAD)
- System of Systems (SOS)
- Technology Obsolescence
- Summary

DISTRIBUTION A. Approved for public release; distribution

TECHNOLOGY OBSOLESCENCE:
ELEMENTS OF THE PROBLEM

- **Product Availability:**
  - **Commercial Life Cycles << Weapon System Development & Operational Lives**
  - **Every Inventory Weapon System Has Serious Obsolescence Problems**
  - **Virtually All Legacy Systems Experienced Nonavailable Parts Prior to 1st Flight**
- **COTS Survivability In Military Environments**
- **Rapid Change in Software Languages, Tools & Technologies**
- **Business Issues, e.g., Prime Contractor Wrap for CFE**
- **Acquisition Issues - Program Strategies & Costs To Mitigate Obsolescence (Open Systems is an up-front investment)**
- **Growth & Upgrading - Capability for Selective Modifications**
- **Budget Issues - ROI Requirements for "ilities" Mods; Multiyear Upgrade Programs; "Color of Money;" etc.**
- **Organic Depot vs. Contractor Responsibility**
- **Multi-Service/Multi-National Configuration Management**
- **Optimizing Individual Systems vs. Common Solutions**
- **Life Cycle Cost Dominated by Software & Non-Digital Hardware**

DISTRIBUTION A. Approved for public release; distribution

TECHNOLOGY OBSOLESCENCE:
BASIC APPROACH TO MITIGATION

- **Must Address Two Distinct Aspects of the Problem:**
  - ***Affordability*** **in Development, Acquisition, & Logistics Support**
  - ***Availability*** **to Maintain Readiness & Operational Support**
- **Ultimate Goal:**
  - **Technology-Independent Design Capture**
  - **Implementation with Available Products When Needed**
  - **Maximum Use of CAE for Specification Compliance Testing, Design Verification, System Integration, etc**
- **Continuing R&D Throughout Life of Program**
  - **Evaluate and Demonstrate Applicability of New Technologies**
  - **Analyze & Correct Performance & Supportability Shortfalls**
  - **Capitalize on Up-Front Investment In Open Systems**
  - **In short, *Do* Evolutionary Acquisition**

DISTRIBUTION A. Approved for public release; distribution

TECHNOLOGY OBSOLESCENCE: *BASIC APPROACH TO MITIGATION* (CONTINUED)

- **Traditional Approaches (Useful In Specific Instances, *Not* a Preferred Solution):**
  - **Lifetime Parts Buys**
  - **Reverse Engineering for Suitable Substitutions**
  - **P^3I & Modification/Replacement of Unsupportable Systems/Subsystems**
- **Architectural Strategies - Open Systems, Clean Partitioning & Domain-Confined Timing**
- **Executable Design Specifications:**
  - **Separate Function from Point Design**
  - **CEENSS (Continuous Electronics Enhancement using Simulatable Specifications)**

DISTRIBUTION A. Approved for public release; distribution

TOPICS

- Context For JSF Avionics Architecture
- Open System Principles for Embedded Real Time Processing
- JSF Avionics Architecture Definition (JAAD)
- System of Systems (SOS)
- Technology Obsolescence
- Summary

DISTRIBUTION A. Approved for public release; distribution

SUMMARY:
ELEMENTS OF A STRATEGY

- **Specify Architectural Attributes, Not Design Specifics**
- **Require Offerors to Describe Approaches/Methodologies:**
  - **System & Software Engineering/Integration/Test**
  - **Compliance/Certification for Applicable Attributes**
  - **Subcontractor Management/Incentives for System-Level Optimization**
- **Use Executable Specifications - Capture Design & Functionality Unambiguously as Simulation Objects**
- **Enforce Rigorous Modular Decomposition with Traceability Up and Down the Hierarchy**
- **Ensure Functional Definitions Are Complete and Technology-Independent**
- **Selectively Apply COTS & Commercial Standards**
- **Use Results of Early Virtual Engineering As Foundation for System Integration Lab For Life of System**

DISTRIBUTION A. Approved for public release; distribution

SUMMARY:
OPEN SYSTEMS IS KEY TO JSF ACQUISITION STRATEGY

- **Physical Integration, Functional Federation Is Essential to Performance, Mission Reliability, Supportability & Affordability**
- **Fighter Avionics Shares Many of the Same Challenges As Those Seen in C^4ISR Systems**
- **Open Systems is a Key Enabler for JSF To Operate In the DoD "System of Systems" Context**
- **Open Systems Enables Ease of Upgrade, Reduced Testing, and Long Term Viability via Modularity**
- **COTS-Based Components and Standards Can be Major Contributors to Mitigating Technology Obsolescence**
- **The JAAD Presents a Framework for Avionics Concept Refinement, Technology Maturation, Conformity to DoD Architectural Standards & Development of Building Codes:**
  - **General Principles & Attributes**
  - **Overall Structure & Context for Application of Standards**

DISTRIBUTION A. Approved for public release; distribution

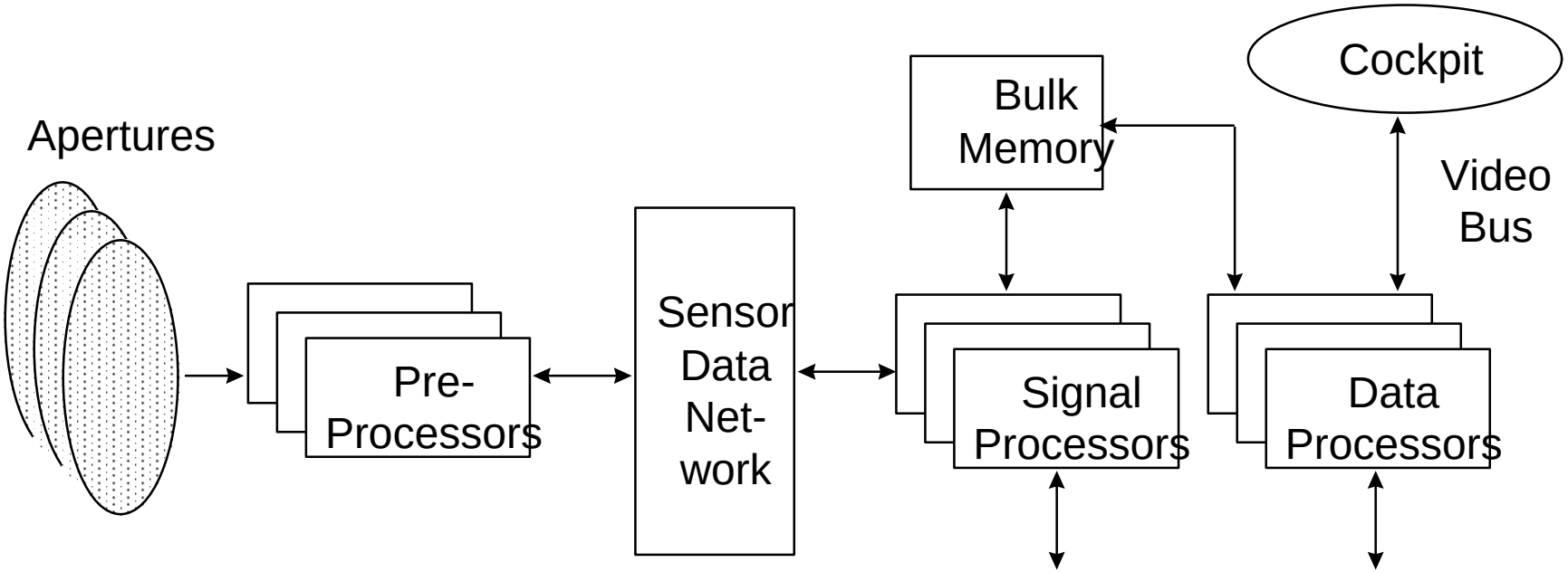
JOINT STRIKE FIGHTER

The Next Generation Strike Fighter

Backups

DISTRIBUTION A. Approved for public release; distribution

PAVE PILLAR

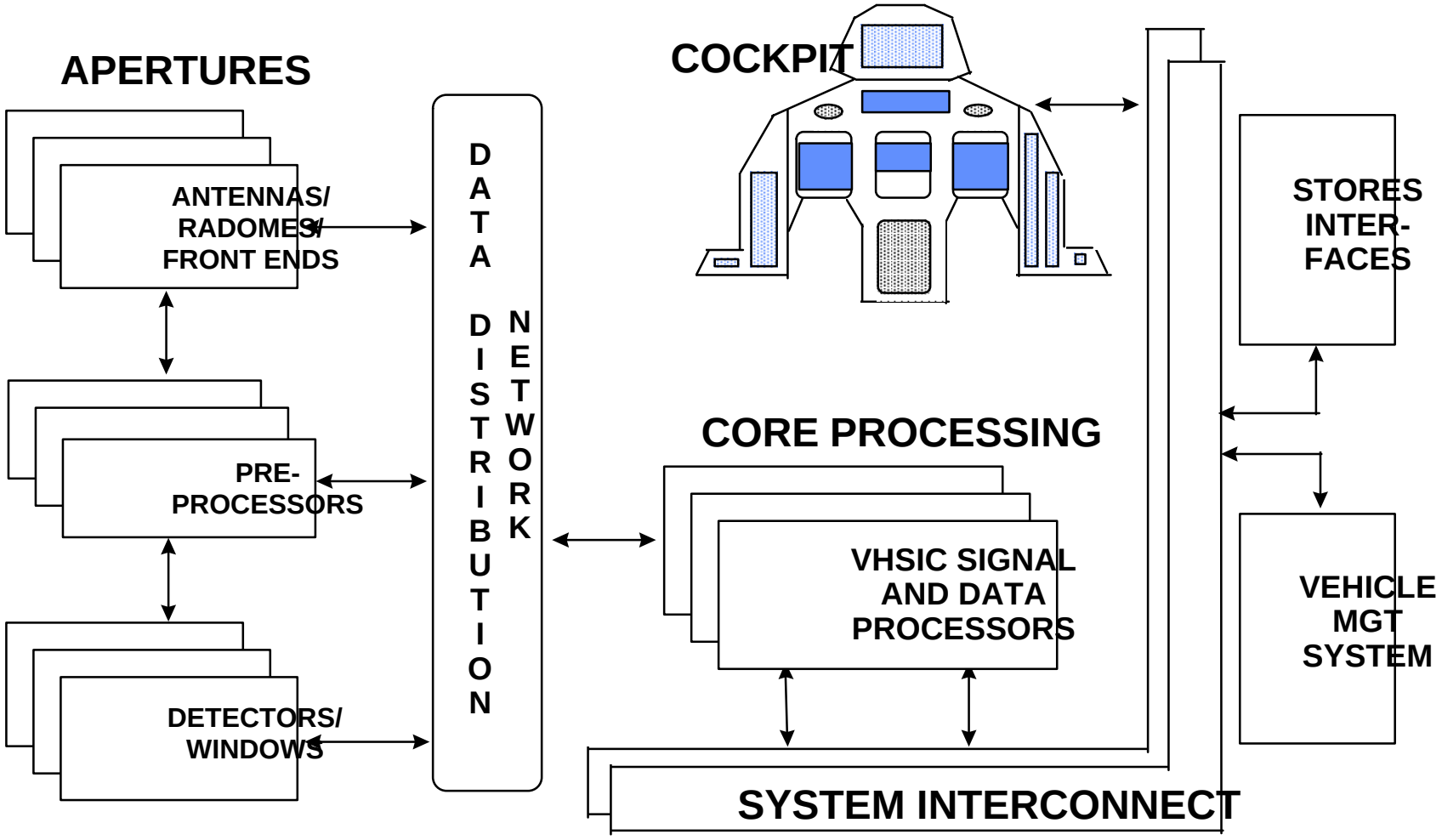


Other Interfaces:

- MIL-STD 1553B
- Avionics - Vehicle Management System
- Avionics - Stores Management System

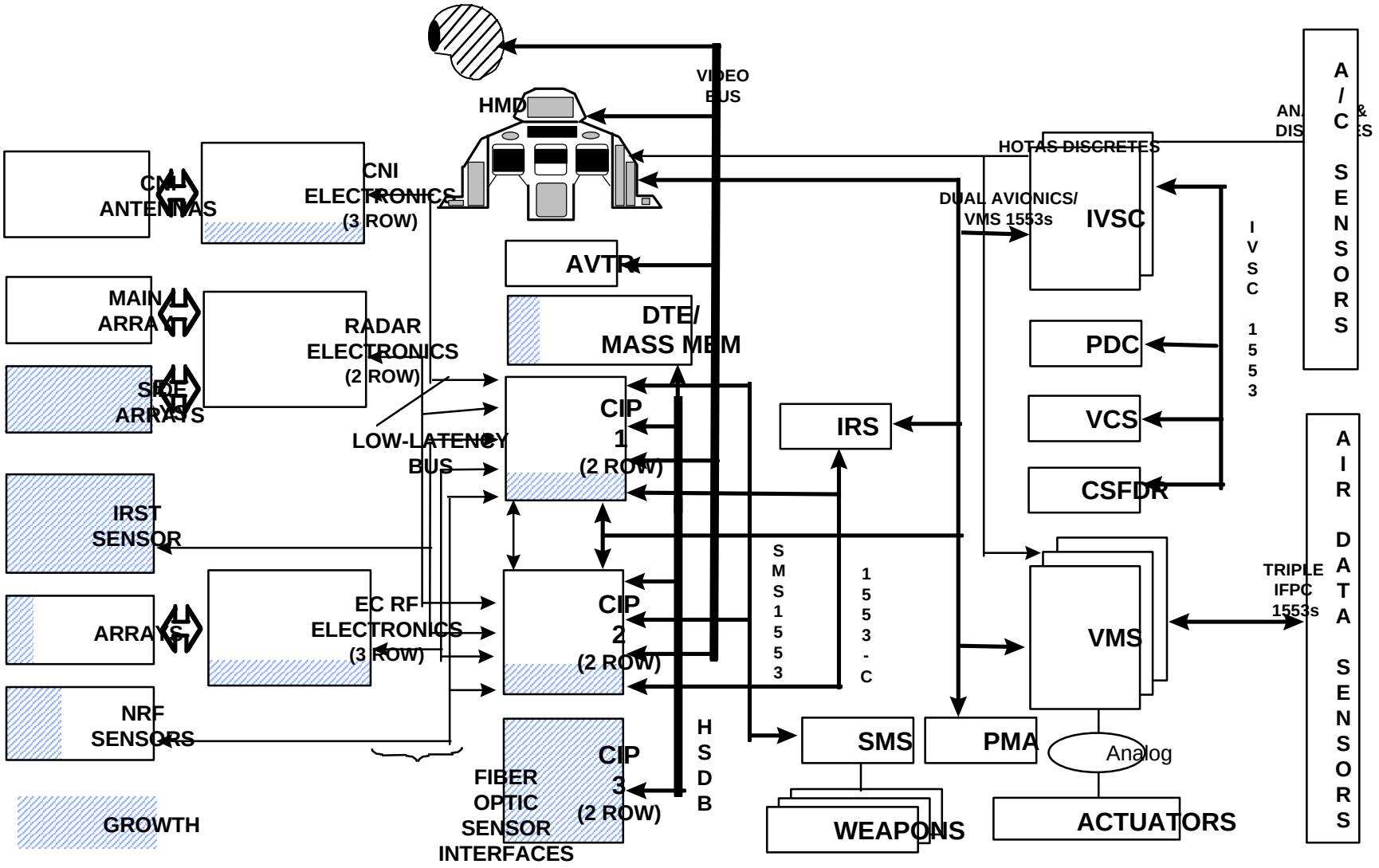
DISTRIBUTION A. Approved for public release; distribution

JIAWG ADVANCED AVIONICS ARCHITECTURE



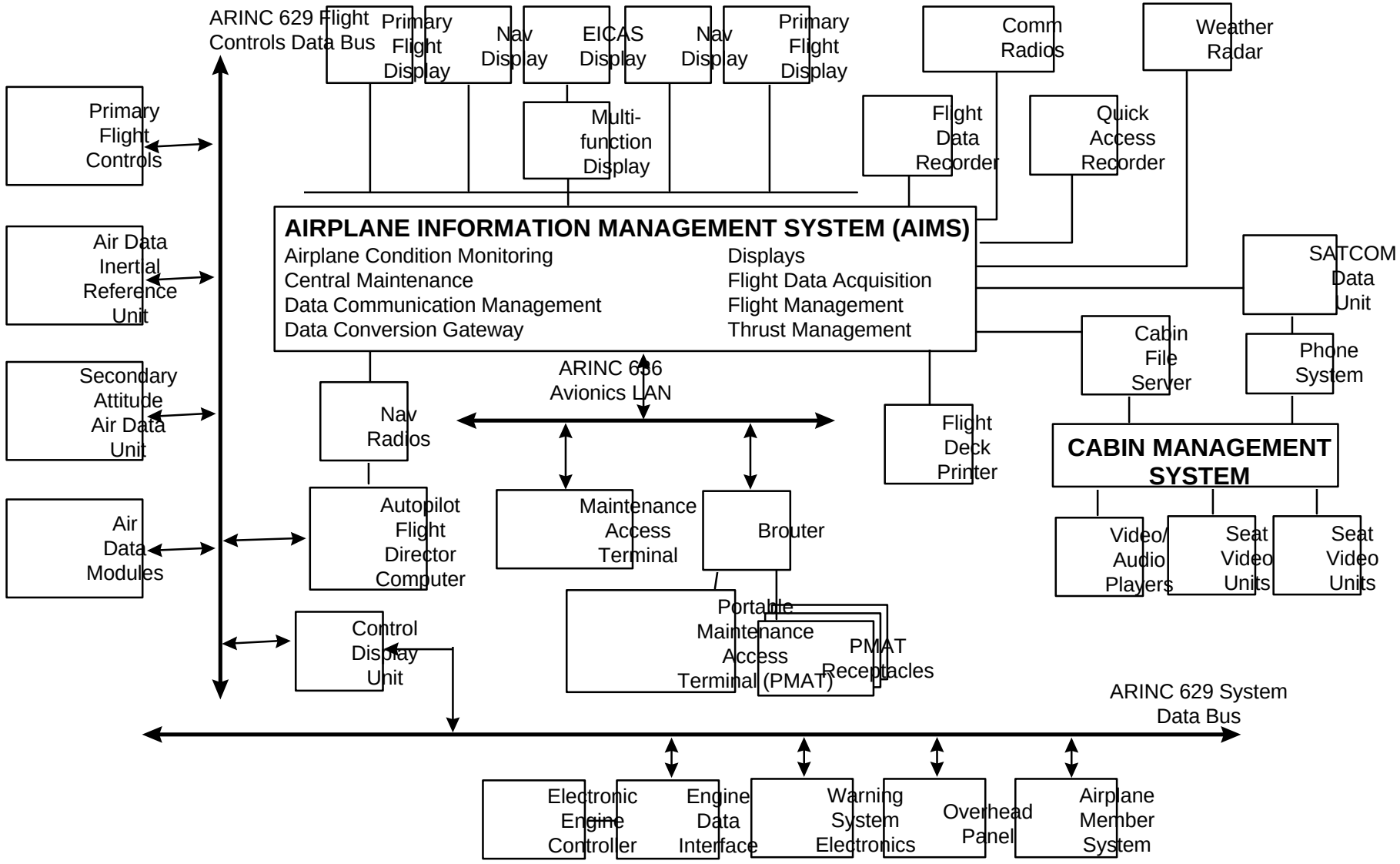
DISTRIBUTION A. Approved for public release; distribution

F-22 ARCHITECTURE



DISTRIBUTION A. Approved for public release; distribution

B777 SYSTEM ARCHITECTURE



DISTRIBUTION A. Approved for public release; distribution

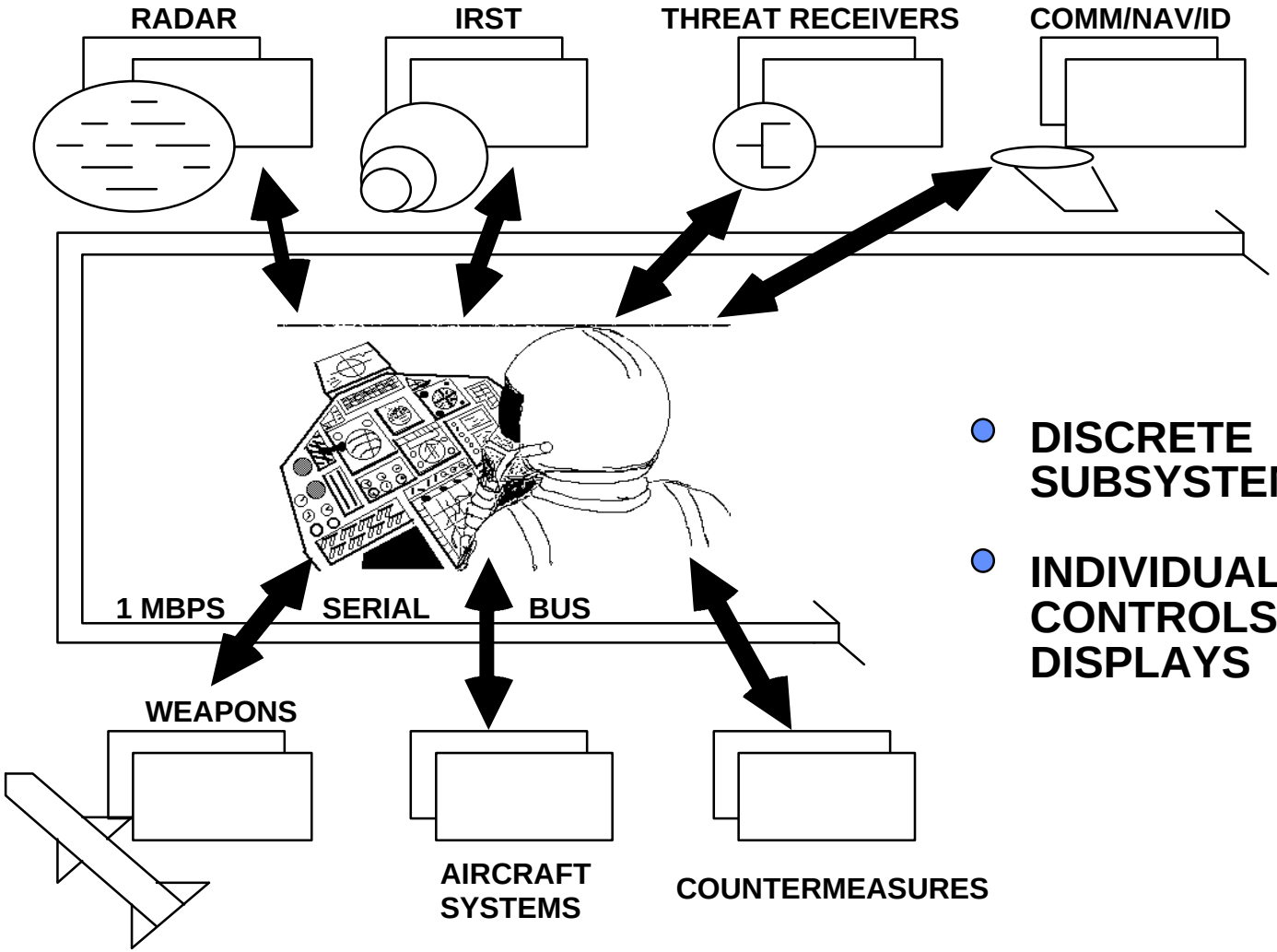
OPEN SYSTEM PRINCIPLES: *SOME KEY TERMS & CONCEPTS* (CONTINUED)

- **Architecture:**
  - **Categories in the Joint Technical Architecture (JTA):**
    - **Operational Architecture - User Requirements**
    - **System Architecture - Three Elements:**
      - **System Components or Resources**
      - **Interactions/Relationships Among Resources**
      - **Global Rules for Design, Operation, & Evolution**
    - **Technical Architecture - Services, Interfaces, Standards & Other Rules & Conventions**

    JAAD
  - **Common Operating Environment (COE) Under the Defense Information Infrastructure (DII) - Defines an Execution Environment & Set of System Services for Use by Applications**
  - **Technical Reference Model (TRM):**
    - **Establishes the Structure of and Relationships Among Software Entities - Applications, Execution Platform, & Interfaces**
    - **These Functions Must Be Mapped Onto a Hardware Architecture**

DISTRIBUTION A. Approved for public release; distribution

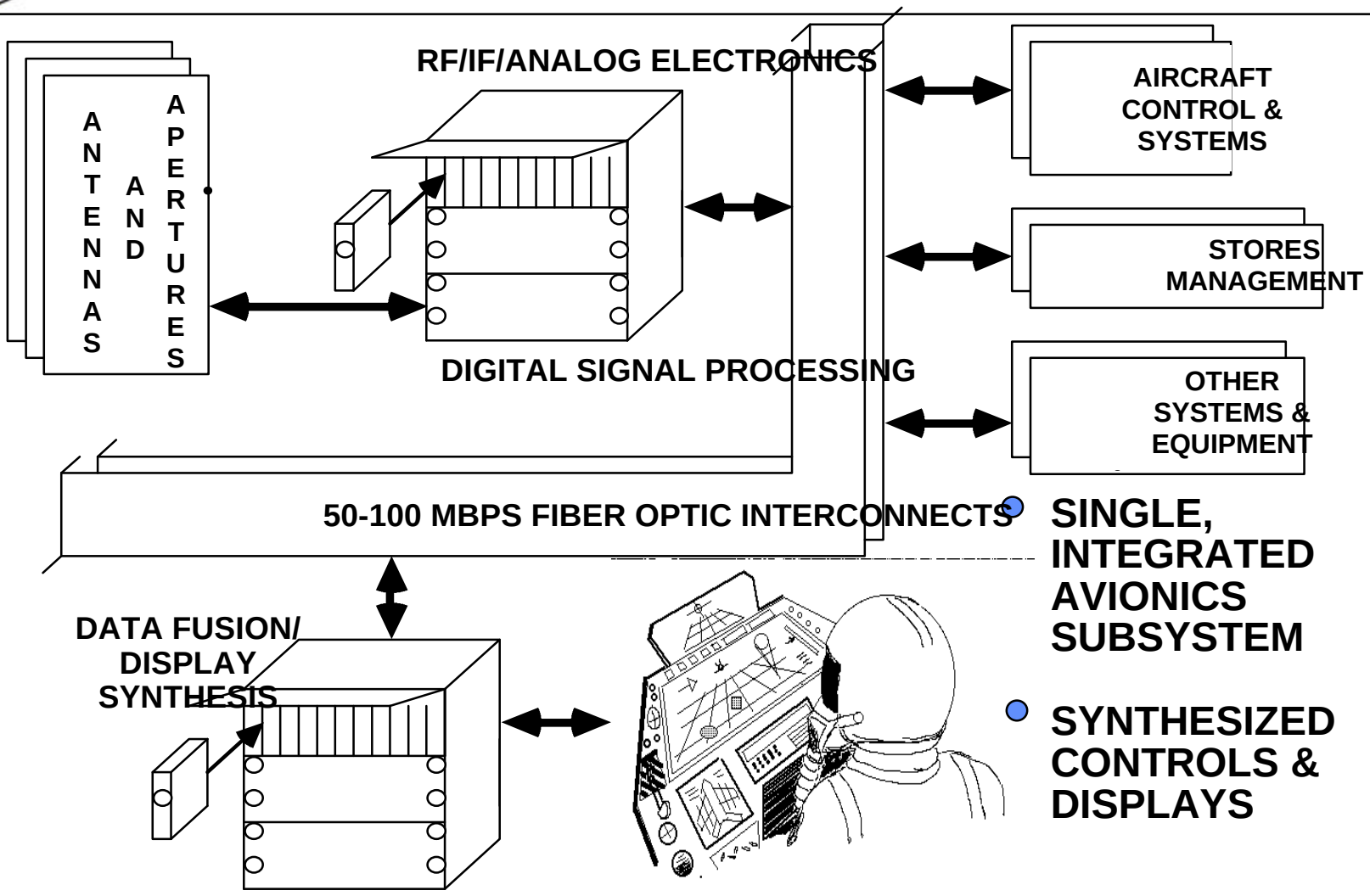
FEDERATED AVIONICS



- DISCRETE SUBSYSTEMS
- INDIVIDUAL CONTROLS & DISPLAYS

DISTRIBUTION A. Approved for public release; distribution

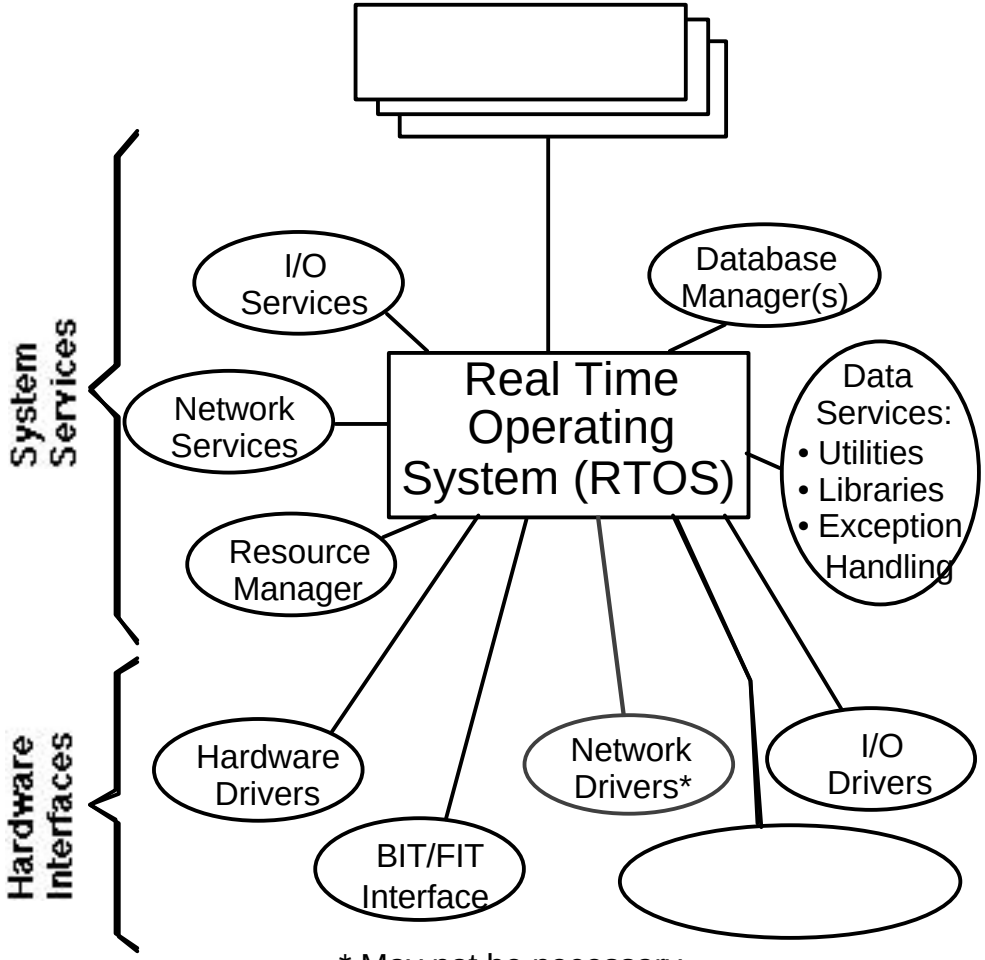
INTEGRATED AVIONICS



- SINGLE, INTEGRATED AVIONICS SUBSYSTEM
- SYNTHESIZED CONTROLS & DISPLAYS

DISTRIBUTION A. Approved for public release; distribution

JAAD SOFTWARE PARTITIONING



* May not be necessary.

DISTRIBUTION A. Approved for public release; distribution